Supplementary Materials

A MULTI-FACETED EVALUATION CRITERIA

The detailed definitions of the multi-faceted evaluation criteria are provided in Tab. 11. The original text is in Chinese, and we translate it into English to align with the language of this paper.

B INSTRUCTION PROMPT

B.1 Graph-based Query Decomposition

Prompt for the graph-based query decomposition is provided in Tab. 12.

B.2 Answer Generation

Prompt for answer generation is provided in Tab. 13.

B.3 LLM Evaluation

Tab. 14 presents the prompt used to instruct the LLM to evaluate the generated text based on multi-faceted criteria (see Tab. 11). To reduce task complexity and enhance evaluation quality, we assess one facet per evaluation and fill the metric title and definition accordingly.

B.4 Built-In Citation

Prompt for entity extraction is provided in Tab. 15. Prompt for citation identification is provided in Tab. 16.

C RETRIEVAL DOCUMENTS FILTERING RULES

HTML Content Filtering. Non-content HTML tags, such as <script> and <style>, are removed using the lxml library to parse the original HTML into a DOM tree. Using XPath selectors, we retain only essential content tags like , <p>, and <article>, ensuring compatibility with irregular HTML structures.

Text Processing. : Extracted text blocks are separated by spaces or line breaks to improve readability. Redundant whitespace and excessive line breaks are removed while preserving paragraph structure. Distracting elements like "Read More," "Click to Continue," or inline emojis are filtered out. Special characters, stopwords (e.g., from publicly available resources like Stopwords JSON[1]), emoji patterns (via regular expressions targeting Unicode Emoji ranges), and irrelevant newline characters are eliminated.

Sensitive Information Filtering. Personal identifiers, such as phone numbers, email addresses, and platform-specific markers are detected and removed.

Text Normalization. Punctuation is standardized to half-width characters, and numbers in various formats are converted to standardized half-width Arabic numerals.

Table 11: Multi-faceted evaluation criteria.

```
(1) Conciseness:
    - The response should directly address the user's
        question.
    - Avoid irrelevant content, unnecessary information,
        or roundabout explanations.
    - Deduct 1 point for each irrelevant statement.

(2) Numerical Precision:
    - If a question requires a specific number, avoid
        vague terms like "several" or "many times."
    - Responses should be precise and specific.
    - Deduct 1 point for each ambiguous statement.

(3) Relevance:
    - If the question specifies constraints (e.g., time,
        location, person, event),
      the answer must adhere to them.
    - Deduct 1 point for each instance of misalignment
        with the question's constraints.

(4) Factuality:
    - The information must be factually correct,
        especially for numerical or factual questions.
    - Deduct 1 point for each incorrect numerical or
        factual statement.

(5) Timeliness:
    - For ongoing news or urgent reports, ensure
        information reflects the latest updates.
    - The current date is {to be filled}.
    - If the question is not time-sensitive, no points
        are deducted.
    - For time-sensitive questions, deduct points
        proportionally based on outdatedness.

(6) Comprehensiveness:
    - The response should comprehensively cover all
        aspects of the user's inquiry.
    - The user should not need further search to grasp
        the full context.
    - Deduct 1 point for each missing essential element.

(7) Clarity:
    - The response should be easy to understand, well-
        structured, and formatted logically.
    - Example: Chronological events should be presented
        in chronological order.
    - Deduct 1 point for unclear or disorganized
        presentation.

(8) Coherence:
    - The response should be logically consistent, with
        smooth transitions between sentences.
    - Deduct 1 point for each instance of incoherent or
        disjointed phrasing.

(9) Insightfulness:
    - The response should provide insightful or unique
        perspectives.
    - Base score: 6 points.
    - Award 0.5-1 additional points for each innovative
        idea or expression.
```

[1]https://github.com/6/stopwords-json/blob/master/dist/ca.json

Table 12: Prompt for graph-based query decomposition.

```
Please analyze the following query and return the
explanation in dictionary format.

Response format:
{'is_complex': True/False, 'sub_queries': [], 'parent_child': []}

Analysis Steps and Principles:

1. **Classify the nature of the query**
   - The query can be classified into one of two types:
     (a) A "complex query" that consists of multiple sub-queries.
     (b) A "simple query" that can be directly answered.
   - If the query is classified as "complex," set 'is_complex' to **True**.
   - If the query is "simple," set 'is_complex' to **False**, and leave 'sub_queries' and 'parent_child' as empty lists.

2. **Decomposing a Complex Query**
   - If the query is classified as "complex," break it down into **sub-queries** and store them in the 'sub_queries'
       list.
   - Decomposition principles:
     1) If a query contains multiple **target entities**, split it into multiple sub-queries.
        - Example: *"What are the latest social news and weather news in Shanghai?"*
          - Target entities: *"social news"*, *"weather news"*.
          - Split into: *"What are the latest social news in Shanghai?"* and *"What are the latest weather news in
              Shanghai?"*.
     2) Each sub-query should be **indivisible** and should not require further decomposition.
     3) No duplicate sub-queries.
     4) When referring to **names of people, places, or organizations**, ensure full and precise descriptions.
        - Example: *"What is the area and population of New Jersey, USA?"*
          - Correct split: *"What is the area of New Jersey, USA?"* and *"What is the population of New Jersey, USA?"*.
          - Incorrect split: *"What is the area of New Jersey?"* and *"What is the population of New Jersey?"*.
     5) The total number of sub-queries **should not exceed 6**.

3. **Analyzing Dependencies Between Sub-Queries**
   - If the query is complex, analyze the **dependency relationships** between sub-queries and store them in '
       parent_child'.
   - Example:
     - *"What natural disasters occurred in Indonesia in April?"*
     - *"How long did this natural disaster last?"*
     - The second question **depends** on the first; thus, the first is the *parent*, and the second is the *child*:
       ```json
       {"parent": "What natural disasters occurred in Indonesia in April?",
        "child": "How long did this natural disaster last?"}
       ```
   - Dependency principles:
     1) If sub-queries are **independent**, 'parent_child' remains an empty list.
     2) If the **child question cannot be answered without the parent**, it is a dependent relationship.
        - Example: "What is the latest iPhone model" is the parent node of "What are the specifications of the latest
            iPhone?"
        - The first question must be answered before the second.
     3) Every possible pair of sub-queries should be evaluated for dependency.
        - A query can be both a *parent* and a *child* in different relationships.

Example:
{Few-Shot Examples}

Query: {Query}
Response: \n
```

D FINE-TUNING

D.1 User Intent Understanding

To promote responsible AI behavior and constrain the scope of queries, we fine-tune the model for the query rejection. We define 11 categories of queries that warrant refusal, including: (1) illegal content, (2) ethical violations, (3) privacy breaches, (4) harmful intent, (5) professional consultations, (6) human-AI interactions, (7) misinformation, (8) technical inquiries, (9) academic requests, (10) planning and consulting inquiries, and (11) creative content generation.

To construct a training dataset, we collect a set of seed queries based on open-source datasets: Do-Not-Answer [73], BeaverTails [31] and Safety-Prompts [65]. Additionally, we generate synthetic queries to compensate the imbalance number for each class in the collected dataset. Then we instruct multiple LLMs by given the definition of the 11 categories to classify and output in JSON format as follows:

```
{
“Refusal”: “Yes/No”,
“Category”: “illegal content/ethical violations/. . . ”
}
```

Table 13: Prompt for answer generation.

```
You are an AI assistant named Xinyu, developed by the
Shanghai Algorithm Innovation Research Institute. You are
performing an encyclopedia Q\&A task. Please generate an
answer based on the provided reference materials and
related Q\&A content.

Question: {Sub-Query}

Related Q\&A:
{Ancestor Node 1: Sub-Query}
{Ancestor Node 1: Answer}
{Ancestor Node 2: Sub-Query}
{Ancestor Node 2: Answer}
...

Reference materials:
{Retrieved Passage 1}
{Retrieved Passage 2}
...

When generating your answer, follow these guidelines:

[Structural Requirements]:
To ensure clarity and organization, you may use one or
more of the following structured formats:
 - **Introduction-Body-Summary**: Introduce the topic,
     elaborate, and summarize key points.
 - **Paragraphs by Subquestion**: Address each
     subquestion in a separate paragraph.
 - **Cause and Effect**: Explain the causes and
     consequences of an event.
 - **Comparison and Contrast**: Describe and compare two
     or more concepts.
 - **Chronological Order**: Describe events or steps in
     order of occurrence.
 - **Problem-Solution**: Introduce a problem and explain
     solutions or strategies.
 - **Pros and Cons**: List the positive and negative
     aspects of a decision or choice.
 - **Definition and Examples**: Provide a definition and
     illustrate it with examples.
 - **Logical Reasoning**: Derive conclusions based on
     assumptions or premises.
 - **List Structure**: Enumerate facts or features for
     easy readability.
 - **Categorization**: Introduce a concept, group it by
     categories, and explain in detail.
 - **Theme and Variations**: Explore a core theme and its
      variations.
 - **Case Study**: Explain a theory or concept through
     specific cases.
 - **Hierarchical Structure**: Arrange information by
     importance or sequence.
 - **Issue and Counterarguments**: Present an issue with
     supporting and opposing views.

[Language Requirements]:
(1) Use concise and clear language.
(2) Ensure that the answer's structure enhances clarity
     and readability.
(3) The response must directly and accurately address the
     question, avoiding irrelevant content.
(4) When citing reference materials, ignore template
     formatting or improper phrasing.
(5) If detailed elaboration is required, output the
     answer in a structured **Markdown** format.

Your Answer: \n
```

Decisions are aggregated using a majority voting mechanism to establish a consensus. Finally, human experts review the results to correct misclassification.

Table 14: Prompt for multi-faceted evaluation.

```
Assume you are an article quality inspector. Please
evaluate the response based on {Metric Title}.
I will provide the user's question and the final response
The maximum score is 10 points, and the scoring rules are
as follows:

{Metric Definition}

Please strictly follow the scoring rules. Example output
format:

'{
    "Issues Identified": "X",
    "Calculation Process": "10-1.0-1.0-1.0 = 7.0",
    "Score": 7
}'

{Few-Shot Examples}

Your final score: \n"
```

Table 15: Prompt for entity extraction.

```
Read the given sentence and extract the contained
information about time, location, persons, and job titles
     .

Your extraction result should be returned in JSON format,
with each field name restricted to one of the following:
["Time", "Location", "Persons", "Job Titles"]

If there are multiple pieces of information of the same
type in the sentence, the corresponding category's value
should be represented as an array.

Below are some examples:

{Few-Shot Examples}

{Sentence}

Extraction result: \n
```

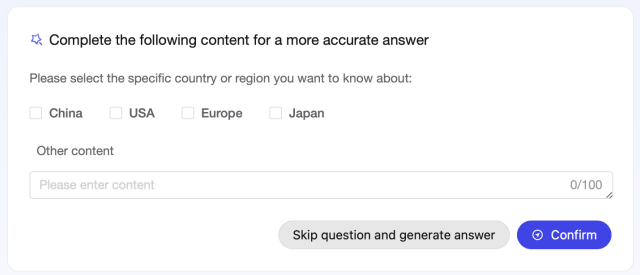

Figure 5: Xinyu's Interface for query disambiguation.

For queries classified as non-refusal, we further prepare a dataset for query disambiguation. We instruct multiple LLMs to analyze whether a query requires further clarification to generate an appropriate response, outputting the results in the following JSON format:

```
{
  "Requires additional input": "Yes/No",
```

Table 16: Prompt for citation identification.

```
You are a journalist skilled in analyzing the correlation
between document information. I will provide you with a
sentence excerpted from a news article, along with
several reference documents used in writing this article.
Your task is to determine which reference document the
excerpted sentence most likely originates from.

The excerpted sentence is:
{Sentence}

The key information contained in this sentence is:
Time: {Time}
Location: {Location}
Person: {Person}
Job Title: {Job Title}
Numbers: {Numbers}

The reference documents used for writing this article and
their respective key information are as follows:
[1] {Retrieved Document}
[2] {Retrieved Document}
[3] ...

When making your determination, ensure that the selected
reference document matches as much key information from
the excerpted sentence as possible. The higher the degree
of key information overlap, the more likely the reference
document is the source of the excerpted sentence.

Your response should contain only a one- or two-digit
number representing the corresponding reference document
number, such as "[2]", "[9]", or "[13]". If you believe
that none of the reference documents are relevant to the
given sentence, return "-1".

The most likely source document number is: \n
```

```
  "Additional options": {
    "Prompt description": "Please select...",
    "Choices": ["xx", "xx", ...]
  }
}
```

Additional options are prepared to present clarifying options for the user as shown in Fig. 5. Again, we use a majority voting mechanism to determine whether additional input is required. This process results in 7K data points. Human experts then review the results, refine the choices, and select high-quality samples. Ultimately, we construct a dataset of 5K data points, where 1/4 of the queries require additional input.

Two models are fine-tuned for the query rejection and query disambiguation tasks using their respective datasets, following Eq. (1).

D.2 Reranker Model

We construct a dataset comprising question-answer pairs using recent real-world data and public datasets such as MS MARCO [11]. To generate hard negatives for fine-tuning, we apply various chunking strategies to create multiple candidate samples resembling the positive examples. These candidates are then ranked based on a base reranker model, selecting the top-300 samples. Next, we leverage multiple LLMs to assess whether each generated sample can answer the corresponding question. If the majority vote is negative, the sample is designated as a hard negative. In total, we generate 56K pairs. Finally, human experts review the results and curate a high-quality subset, retaining 13K pairs for fine-tuning.

Table 17: Comparison of answers generated with or without GQD.

```
Query: "Analyze how 'Blue Moon' and 'Super Moon' have
     evolved in cultural and folk traditions throughout
     history."

Answer with GQD:
In ancient times, a 'Blue Moon' was often seen as an
ominous sign linked to disasters, reflecting early
superstition. Over centuries, it evolved from referring
to the fourth full moon in one season to the second full
moon in a calendar month-highlighting changes in
calendars and beliefs. Likewise, early Chinese traditions
revered a Super Moon as a symbol of cosmic harmony,
whereas modern science explains it simply as a near-
perigee full moon. This trajectory captures a shift from
mystical interpretation to scientific understanding in
various cultures...

Answer without GQD:
Historically, 'Blue Moon' and 'Super Moon' did not draw
much cultural attention; ancient societies were more
generally focused on lunar cycles. These terms gained
modern popularity mainly due to social media, prompting
renewed public interest.
```

D.3 Built-In Citation

As this module requires high efficiency, we fine-tune two SLMs (Qwen2.5-3B), using its larger counterpart, Qwen2.5-72B. We collect a set of passages and use Qwen2.5-72B to extract the entities from each sentence based on the prompt provided in systems Supplementary Material B.4. If any entities can be extracted, we retain the (passage, sentence, entities) triplet as part of the dataset. This process yields 33K data points. Human experts then review the data to correct errors and remove low-quality samples, ultimately retaining 26K data points. We fine-tune the entity extraction SLM using sentences paired with their corresponding entities. Additionally, we fine-tune another SLM to retrieve the relevant passage from a given set based on the extracted entities, using the prompt provided in Supplementary Material B.4.

E ADDITIONAL ABLATION RESULTS

Multi-Faceted Evaluation by LLM. Tab. 18 presents the results of a multi-faceted evaluation conducted by GPT-4O. While the absolute values differ from those obtained through human evaluation (see Tab. 1), the rankings remain similar, demonstrating a strong correlation (see Tab. 2). These results confirm that LLM-based evaluation is also indicative of performance.

Table 18: Multi-faceted comparison of different approaches based on GPT-4O. Higher value indicates better performance, 10 is the maximum.

Model	Conciseness	Numerical Precision	Relevance	Factuality	Timeliness	Comprehensiveness	Clarity	Coherence	Insightfulness	Average
Perplexity AI	**9.913**	**9.607**	**9.740**	9.727	8.120	8.280	9.887	**9.853**	6.613	9.082
Tiangong AI [6]	9.819	9.188	9.570	9.738	7.758	7.517	9.839	9.799	6.161	8.821
Ernie Bot [2]	9.814	9.152	9.556	9.648	8.062	7.924	9.745	9.814	6.552	8.918
KIMI [4]	9.695	9.359	9.576	9.675	8.059	8.305	9.686	9.720	6.432	8.945
Metaso [50]	9.781	8.932	9.493	9.596	7.589	6.842	9.712	9.589	5.801	8.593
ChatGLM [16]	9.733	9.274	9.568	**9.745**	7.986	7.911	9.863	9.808	6.603	8.943
Baichuan [1]	9.433	9.053	9.307	9.403	7.813	7.832	9.373	9.200	6.640	8.673
Tongyi [7]	9.747	8.900	9.313	9.527	7.700	7.940	9.827	9.740	6.493	8.799
Xinyu (Ours)	9.880	9.547	9.547	9.731	**8.300**	**8.533**	**9.900**	9.747	**7.107**	**9.144**